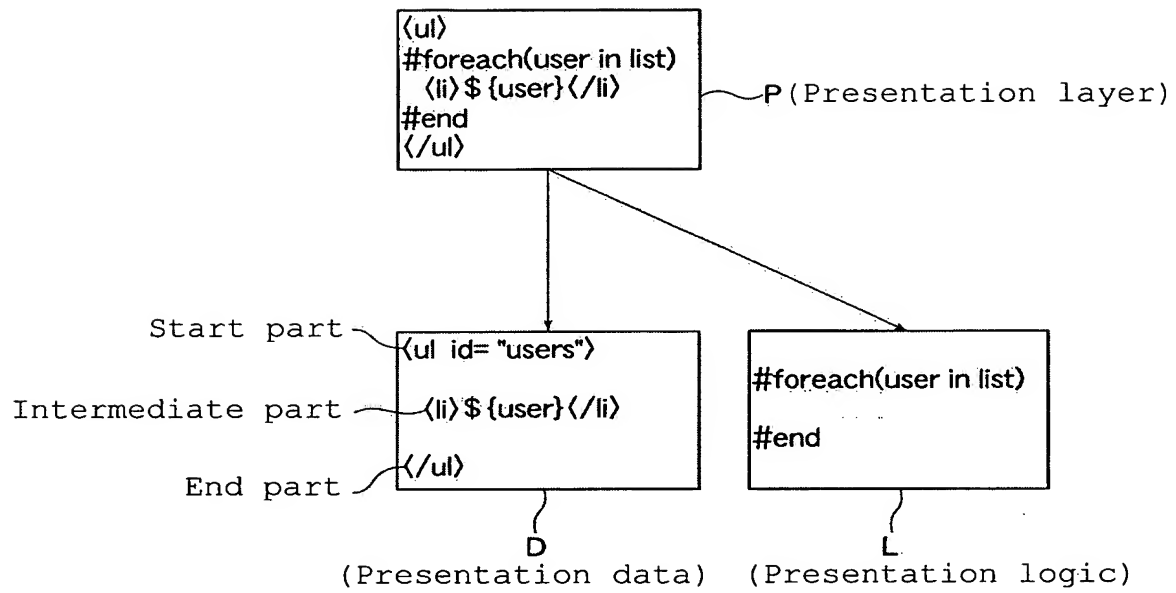
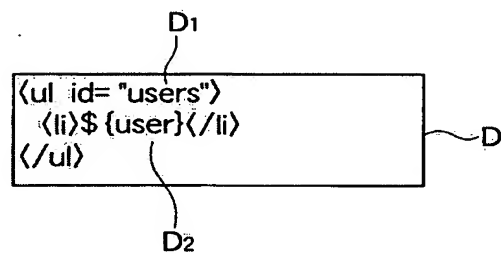


[FIG. 1]



[FIG. 2]



[FIG. 3]

The diagram shows a series of macro definitions and an expansion, grouped by curly braces and labeled M1 through M5 on the right side. The code is as follows:

```
#defmacro(head_users)
  <ul>
#end

#defmacro(body_users)
  <li>${user}</li>
#end

#defmacro(foot_users)
  </ul>
#end

#defmacro(elem_users)
  #expand(head_users)
  #expand(body_users)
  #expand(foot_users)
#end

#expand(elem_users)
```

Labels on the right:

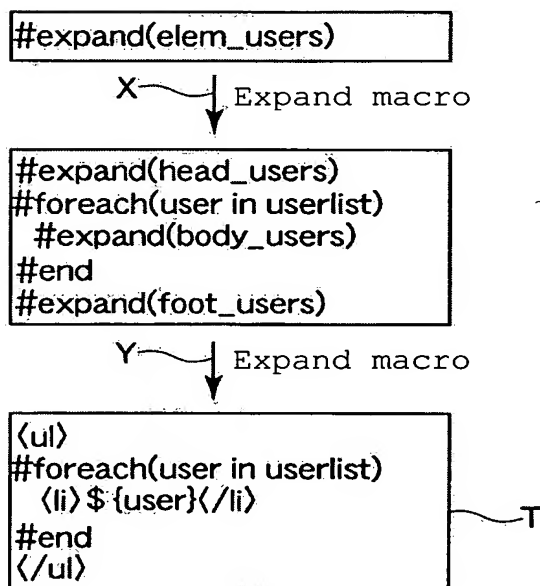
- M1: points to the `#defmacro(head_users)` block.
- M2: points to the `#defmacro(body_users)` block.
- M3: points to the `#defmacro(foot_users)` block.
- M4: points to the `#defmacro(elem_users)` block.
- M5: points to the `#expand(elem_users)` line.

[FIG. 4]

```
#defmacro(elem_users)
#expand(head_users)
#foreach(user in userlist)
#expand(body_users)
#end
#expand(foot_users)
#end
```

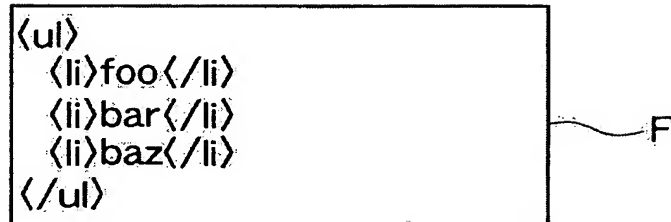
L

[FIG. 5]



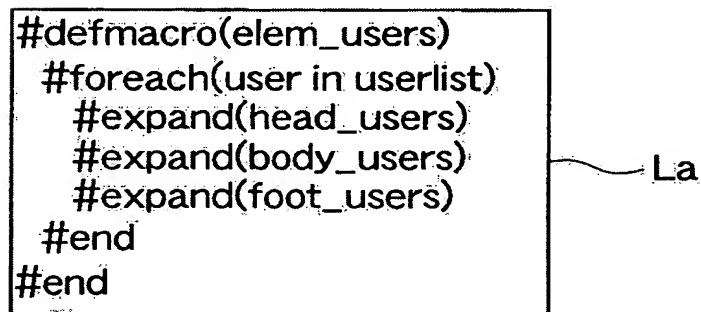
[FIG. 6]

```
<ul>  
  <li>foo</li>  
  <li>bar</li>  
  <li>baz</li>  
</ul>
```



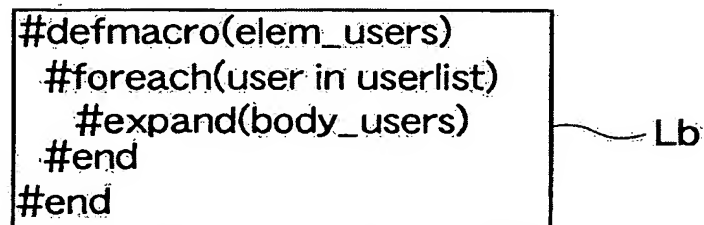
[FIG. 7]

```
#defmacro(elem_users)  
  #foreach(user in userlist)  
    #expand(head_users)  
    #expand(body_users)  
    #expand(foot_users)  
  #end  
#end
```



[FIG. 8]

```
#defmacro(elem_users)  
  #foreach(user in userlist)  
    #expand(body_users)  
  #end  
#end
```



[FIG. 9]

```
#defmacro(elem_users)
  #set(i = 0)
  #foreach(user in userlist)
    #set(i = i+1)
    #if(i%2 ==0)
      #set(color = " red ")
    #else
      #set(color = " blue ")
    #end
    #expand(head_users)
    #expand(body_users)
    #expand(foot_users)
  #end
#end
```

Lc

[FIG. 10]

```
class Example {  
    variable user;  
  
    method generate() {  
        doc = new Document();  
        addElementFoo(doc);  
        return doc.transformToXmlString();  
    }  
  
    method addElementUsers(Element parent) {  
        elem = new Element("ul");  
        addContentUsers(elem);  
        parent.appendChild(elem);  
        return elem;  
    }  
  
    method addContentUsers(Element parent) {  
        elem = new Element("li");  
        text = new Text(user);  
        elem.appendChild(text);  
        parent.appendChild(elem);  
        return elem;  
    }  
}
```

[FIG. 11]

```
class MyExample extends Example {  
  variable userlist;  
  
  method addConentUsers(Element parent) {  
    foreach (user in userlist) {  
      super.addContentUsers(parent);  
    }  
  }  
}
```

[FIG. 12]

```
class MyExample extends Example {  
  variable user;  
  variable userlist;  
  
  method addElementUsers(Element parent) {  
    foreach (user in userlist) {  
      super.addElementUsers(parent);  
    }  
  }  
}
```

[FIG. 13]

```
class MyExample extends Example {  
    variable userList;  
  
    method addElementUsers(Element parent) {  
        super.addContentUsers(parent);  
    }  
}
```

[FIG. 14]

```
class MyExample extends Example {  
    variable userList;  
  
    method addElementUsers(Element parent) {  
        i = 0;  
        foreach (user in userList) {  
            i = i+1;  
            if (i % 2 == 0 )  
                color = " red " ;  
            else  
                color = " blue " ;  
            super.addElementUsers(parent);  
        }  
    }  
}
```


[FIG. 15]

```
<ul>  
#foreach(user in list)  
  <li>${user}</li>  
#end  
</ul>
```

Conventional template
(Presentation data + Presentation logic)